UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/038,351	12/31/2001	Joseph P. Bratt	04860.P2686	7732

7590 03/09/2005

James C. Scheller
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard, Seventh Floor
Los Angeles, CA 90025-1026

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 03/09/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/038,351	BRATT ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 December 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35,37-41,43-56 and 61-85* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35,37-41,43-56 and 61-85* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *13 December 2004*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-35, 37-41, 43-56, and 61-79 and new claims 80-85 have been examined. Claims 2, 4, 12, 13-15, 17. 19. 20, 22, 23, 37, 48, 56, 67, 68, 70, 72, 74, 76, and 77 have been amended as per Applicant's request. New claims 80-85 have been added as per Applicant's request. Claims 36, 42, and 57-60 have been cancelled as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Amendment as filed 13 December 2004 and IDS as filed 13 December 2004.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

4. The applicant is requested to review the specification and update the status of all co-pending applications made mention of, replacing attorney docket numbers with current U.S. application or patent numbers when appropriate.

Double Patenting

5. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process ... may obtain a patent therefor ..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957); and *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970).

6. A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the conflicting claims so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

7. Claims 14, 16, 18, 23, 26, 27 and 33 are provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1, 4, 5, 1, 4, 5 and 11, respectively, of copending Application No. 10/038,478. This is a provisional double patenting rejection since the conflicting claims have not in fact been patented.

8. Below is a table showing how the limitations from Application No. 10/038,478 map to limitations found in the instant application.

10/038,478	Instant Application
Claim 1	Claim 14
A method for execution by a microprocessor in response to receiving a single instruction	An execution unit in a microprocessor
Receiving a first plurality of numbers and a second plurality of numbers	A first circuit configured to accept a first plurality of numbers; A second circuit configured to accept a second plurality of numbers;
Each of the first plurality of numbers pointing to one of a plurality of entries	Each of the first plurality of numbers pointing to one of a plurality of entries
Each of the plurality of entries being in one of a plurality of look-up tables	A plurality of look-up tables; Each of the plurality of entries being in one of the plurality of look-up tables
Replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers	The third circuit replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers
Wherein the above operations are performed in response to the microprocessor receiving the single instruction	A third circuit coupled to the first circuit, second circuit, and the plurality of look-up tables, in response to the microprocessor receiving a single instruction
Wherein the microprocessor comprises a media processor integrated with a memory controller for host memory on a single integrated circuit	Wherein the microprocessor comprises a media processor integrated with a memory controller for host memory on a single integrated circuit
Claims 4 and 5	Claims 16 and 18
A method for execution by a microprocessor in response to receiving a single instruction	An execution unit in a microprocessor
Replacing at least one entry in at least one	A plurality of look-up tables;

of a plurality of look-up units in a microprocessor unit with at least one number using a Direct Memory Access (DMA) controller	A first circuit coupled to the plurality of look-up tables and a Direct Memory Access (DMA) controller, The first circuit replacing at least one entry in at least one of the plurality of look-up tables with at least one data element using the DMA controller
Wherein the above operations are performed in response to the microprocessor receiving the single instruction	The first circuit, in response to the microprocessor receiving a single instruction
Claim 1	Claim 23
A method for execution by a microprocessor in response to receiving a single instruction	An execution unit in a microprocessor
Receiving a first plurality of number and a second plurality of numbers	Means for receiving a first plurality of numbers and a second plurality of numbers
Each of the first plurality of numbers pointing to one of a plurality of entries	Each of the first plurality of numbers pointing to one of a plurality of entries
Each of the plurality of entries being in one of a plurality of look-up tables	Each of the plurality of entries being in one of a plurality of look-up tables
Replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers	Means for replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers
Wherein the above operations are performed in response to the microprocessor receiving the single instruction	Wherein the above means are performed in response to the microprocessor receiving the single instruction
Wherein the microprocessor comprises a media processor integrated with a memory controller for host memory on a single integrated circuit	Wherein the microprocessor comprises a media processor integrated with a memory controller for host memory on a single integrated circuit
Claims 4 and 5	Claims 26 and 27
A method for execution by a microprocessor in response to receiving a single instruction	An execution unit in a processor
Replacing at least one entry in at least one of a plurality of look-up units in a microprocessor unit with at least one number using a Direct Memory Access (DMA) controller	Means for replacing at least one entry in at least one of a plurality of look-up units in a microprocessor unit with at least one number using a Direct Memory Access (DMA) controller

Wherein the above operations are performed in response to the microprocessor receiving the single instruction	Wherein the above means are performed in response to the microprocessor receiving the single instruction
Claim 11	Claim 33
A method for execution by a microprocessor in response to receiving a single instruction	An execution unit in a microprocessor
Receiving a plurality of numbers	Means for receiving a plurality of numbers
Partitioning look-up memory into a plurality of look-up tables	Means for partitioning look-up memory into a plurality of look-up tables
Looking up simultaneously a plurality of elements from the plurality of look-up tables	Means for looking up simultaneously a plurality of elements from the plurality of look-up tables
Each of the plurality of elements being in one of the plurality of look-up tables and being pointed to by one of the plurality of numbers	Each of the plurality of elements being in one of the plurality of look-up tables and being pointed to by one of the plurality of numbers
Wherein the above operations are performed in response to the microprocessor receiving the single instruction	Wherein the above means are performed in response to the microprocessor receiving the single instruction

9. Applicant is advised that should claims 16 and 18, claims 17 and 19, and claims 26 and 27 be found allowable, claims 18, 19, and 26 will be objected to under 37 CFR 1.75 as being a substantial duplicate thereof. When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper after allowing one claim to object to the other as being a substantial duplicate of the allowed claim. See MPEP § 706.03(k).

Claim Rejections - 35 USC § 112

10. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

11. Claim 73 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claim 73 claims "means for generating an indicator indicating whether *any bit after the last bit of input is used* in obtaining the first result." After perusal of the written description, the Examiner could not locate a description on how a system utilizes bit(s) that do not exist, i.e. bits after the last bit. A last bit is the final bit and there are no more bits after it.

12. Claim 73 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 73 claims "means for generating an indicator indicating whether *any bit after the last bit of input is used* in obtaining the first result." After perusal of the written description, the Examiner could not locate a description on how a system utilizes bit(s) that do not exist, i.e. bits after the last bit. A last bit is the final bit and there are no more bits after it.

Claim Rejections - 35 USC § 102

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this

subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

14. Claims 1-15, 17, 19-25, 33-35, 37-41, 43-56, and 61-85 are rejected under 35 U.S.C. 102(e) as being anticipated by Barry et al., U.S. Patent No. 6,397,324.

15. Regarding claim 1, Barry has taught an execution unit in a microprocessor, the execution unit comprising:

a. Look-up memory (431/433 of Fig.4),

b. A first circuit coupled to the look-up memory,

i. The first circuit, in response to the microprocessor receiving a first instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32), partitioning the look-up memory into a first plurality of look-up tables (see Col.7 lines 54-62 and Col.9 lines 63-67 and Col.11 lines 10-13),

ii. The first circuit, in response to the microprocessor receiving a second instruction (see "L4TBL" on Col.11 lines 32-48), partitioning the look-up memory into a second plurality of look-up tables which are different from the first plurality of look-up tables (see Col.11 lines 33-36). Here, the look-up tables are different in size and number between the L2TBL and L4TBL instructions.

16. Regarding claim 2, Barry has taught an execution unit as in claim 1, wherein:

a. A total number of bits used by each entry in the first plurality of look-up tables is different from a total number of bits used by each entry in the second plurality of look-up tables (see Col.10 line 62 – Col.11 line 48). Here, the "size" field of both the L2TBL and the L4TBL instructions specifies whether the instruction is to use

dual word, dual-half word, or dual byte data is to be used in the look table entries (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28). Thus a L2TBL and a L4TBL instruction can use different number of bits for each entry.

b. The microprocessor is a media processor formed in a monolithic semiconductor substrate, which comprises a memory controller (485 of Fig.4) for controlling host memory (431/433 of Fig.4), said media processor being coupled to said memory controller (see Fig.4). Here, while not taught explicitly, it is inherent that the processor of Barry is formed on a monolithic semiconductor substrate, as that is how digital circuits are created.

17. Regarding claim 3, Barry has taught an execution unit as in claim 1, wherein a total number of entries in each of the first plurality of look-up tables is different from a total number of entries in each of the second plurality of look-up tables (see Col.10 line 62 – Col.11 line 48). Here, Barry supports look-up table sizes of 256 entries for the L4TBL instruction (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries for the L2TBL instruction, each entry being an 8, 16 or 32-bit entry.

18. Regarding claim 4, Barry has taught an execution unit as in claim 1, wherein the look-up memory comprises a plurality of look-up units, and wherein the first circuit is to configure the plurality of look-up units into a third plurality of look-up tables in response to the microprocessor receiving a third instruction (see "LTBL" on Col.10 lines 24-61). Here, the LTBL instruction partitions the look-up memory into a single look-up table (see Col.10 lines 24-61).

19. Regarding claim 5, Barry has taught an execution unit as in claim 4, wherein each of the third plurality of look-up units contains 256 8-bit entries (see Col.10 line 62 - Col.11 line 48). Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry.

20. Regarding claim 6, Barry has taught an execution unit as in claim 4, wherein a total number of entries in each of the third plurality of look-up tables is one of:

a. 256 (see Col.11 line 44),

b. 512,

c. 1024.

21. Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 6.

22. Regarding claim 7, Barry has taught an execution unit as in claim 4, wherein a total number of bits used by each entry in the third plurality of look-up tables is one of:

a. 8 ("two bytes" in Col.12 lines 14-28),

b. 16 ("two halfwords" in Col.12 lines 14-28),

c. 24.

23. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 7.

24. Regarding claim 8, Barry has taught an execution unit as in claim 1, further comprising:

a. A second circuit coupled to the look-up memory, the second circuit configured to receive a plurality of numbers (see An/Rz of Fig.6), in response to the microprocessor receiving the first instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32), the first plurality of look-up tables looking up simultaneously a plurality of entries, each of the plurality of entries being in one of the plurality of look-up tables and being pointed to by one of the plurality of numbers (see Col.9 lines 41-67 and Col.12 lines 14-27). Here, the L2TBL instruction, which is the LTBL instruction modified to perform two look-up table look-ups (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.10 line 62 – Col.11 line 32). The above two pointers received point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

25. Regarding claim 9, Barry has taught an execution unit as in claim 1, further comprising:

a. A second circuit coupled to the look-up memory, the second circuit configured to receive a string of bits, in response to the microprocessor receiving the first instruction (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register address.

b. The second circuit generating a plurality of indices using a plurality of segments of bits in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL

instruction specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two indices to entries into the look-up tables (see Col.10 line 62 – Col.11 line 32).

c. The first plurality of look-up tables (431/433 of Fig.4) looking up simultaneously a plurality of entries each of the plurality of entries being in one of the plurality of look-up tables and being pointed to by one of the plurality of indices (see Col.9 lines 41-67 and Col.12 lines 14-27). Here, the above two pointers received point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

26. Regarding claim 10, Barry has taught an execution unit as in claim 9, further comprising:

a. A third circuit coupled to the look-up memory, the third circuit combining the plurality of entries into a first result (see Col.11 lines 10-32). Here, the two outputted entries from each look-up table are combined and stored in register Rt, with each entry stored in one half of register Rt.

27. Regarding claim 11, Barry has taught an execution unit as in claim 10, further comprising:

a. A forth circuit coupled to the second circuit, the forth circuit configured to receive a plurality of data elements specifying the plurality of segments in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register addresses.

28. Regarding claim 12, Barry has taught an execution unit as in claim 10, further comprising:

a. A fifth circuit coupled to the second circuit, the fifth circuit configured to receive at least one format (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

b. A sixth circuit coupled to the fifth circuit and the third circuit, in response to the microprocessor receiving the first instruction (see Col.11 lines 10-32):

i. The fifth circuit formatting the string bits into a least one escape data using the at least one format (see Col.11 lines 10-32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The output for the look-up tables at these indices is formatted, i.e. sign-extended, if necessary according to the "size" field of the instruction.

ii. The sixth circuit combing the at least one escape data with the first result into a second result (see Col.10 line 62 – Col.11 line 32). Here, the two outputted entries from each look-up table are sign-extended (i.e. combined

with the escape data) if necessary, and then combined and stored in register Rt, with each entry stored in one half of register Rt.

29. Regarding claim 13, Barry has taught a processing system comprising a plurality of execution units including an execution unit as in claim 1 (see Fig.4).

30. Regarding claim 14, Barry has taught an execution unit in a microprocessor (see Fig.4) comprising:

a. A plurality of look-up tables (431/433 of Fig.4),

b. A first circuit configured to accept a first plurality of numbers (see An/Ri of Fig.4 or An/Rz of Fig.8), each of the first plurality of numbers pointing to one of a plurality of entries, each of the plurality of entries being in one of the plurality of look-up tables (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

c. A second circuit configured to accept a second plurality of numbers (see Rs of Fig.4 or Rte/Rto of Fig.8),

d. A third circuit coupled to the first circuit, the second circuit, and the plurality of look-up tables, the third circuit, in response to the microprocessor receiving a single instruction, replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers (see Col.9 lines 41-62 and Col.12 lines 14-27); and

e. Wherein the microprocessor is a media processor integrated with a memory controller (485 of Fig.4) for host memory (431/433 of Fig.4) on a single integrated circuit (see Fig.4).

31. Regarding claim 15, Barry has taught a processing system comprising a plurality of execution units including an execution unit as in claim 14 (see Fig.4).

32. Regarding claim 17, Barry has taught a processing system comprising a plurality of execution units including an execution unit as in claim 16 (see Fig.4).

33. Regarding claim 19, Barry has taught a processing system comprising a plurality of execution units including an execution unit as in claim 18 (see Fig.4).

34. Regarding claim 20, Barry has taught an execution unit in a microprocessor comprising:

a. A plurality of look-up tables (431/433 of Fig.4),

b. A first circuit coupled to the plurality of look-up tables, the first circuit configured to receive a string of bits (see Col.10 line 62 – Col.11 line 32),

c. A second circuit coupled to the plurality of look-up tables and the first circuit, the second circuit configured to receive a plurality of data elements, in response to the microprocessor receiving a single instruction, the second circuit generating a plurality of indices using the plurality of data elements and the string of bits, the

plurality of look-up tables looking up simultaneously a plurality of entries using the plurality of indices (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The above two indices point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file (see Col.9 lines 41-67, Col.11 lines 1-13 and Col.12 lines 14-27).

d. A third circuit coupled to the plurality of look-up tables, the third circuit combining the plurality of entries into a first result (see Col.11 lines 10-32). Here, the two outputted entries from each look-up table are combined and stored in register Rt, with each entry stored in one half of register Rt.

35. Regarding claim 21, Barry has taught an execution unit as in claim 20, further comprising:

a. A fifth circuit coupled to the second circuit, the fifth circuit configured to receive at least one format (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

b. A sixth circuit coupled to the fifth circuit and the third circuit, in response to the microprocessor receiving the single instruction, the fifth circuit formatting the string of bits into at least one escape data using the at least one format, and the sixth circuit combining the at least one escape data with the first result into a second result (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a particular format with the sign-extension bit. The sign-extension bit designates whether the output data is sign-extended or not. The instruction retrieves data into the Rt register, and, depending on the sign-extension bit, sign-extends the data or not. The instruction only retrieves half-word data which is loaded into the H0 portion of the register. The sign-extension bit determines whether a string of bits needs to be formatted for the H1 portion of the registers, i.e. formatting escape data. The H1 portion is then combined with the H0 portion to make a sign-extended result in Rz (Barry column 11, lines 15-27).

36. Regarding claim 22, Barry has taught a processing system comprising a plurality of execution units including an execution unit as in claim 21 (see Fig.4).

37. Regarding claim 23, Barry has taught an execution unit in a microprocessor (see Fig.4), the execution unit comprising:

a. Means for receiving a first plurality of numbers (see An/Ri of Fig.4 or An/Rz of Fig.8) and a second plurality of numbers (see Rs of Fig.4 or Rte/Rto of Fig.8), each of the first plurality of numbers pointing to one of a plurality of entries, each of the plurality of entries being in one of a plurality of look-up tables (431/433 of Fig.4) (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the

S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

b. Means for replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers (see Col.9 lines 41-62 and Col.12 lines 14-27),

c. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "S2TBL" on Col.12 lines 13-27); and

d. Wherein the microprocessor is a media processor integrated with a memory controller (485 of Fig.4) for host memory (431/433 of Fig.4) on a single integrated circuit (see Fig.4).

38. Regarding claim 24, Barry has taught an execution unit as in claim 23, wherein the first plurality of numbers are received form a first entry in a register file (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27), and the second plurality of numbers are received form a second entry in the register file (see Col.9 line 25 – Col.10 line 20 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction specifies, using even/odd addressing, two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an

entry in one of the look-up tables (see Col.12 lines 14-27). Thus, the execution unit "receives" the first plurality of numbers from a first entry in a register file (An + Rz). Furthermore, the S2TBL instruction specifies two pieces of data denoted by odd and even addresses (each data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

39. Regarding claim 25, Barry has taught an execution unit as in claim 24, wherein the single instruction specifies indices of the first (An and Rz of Fig.8, see also Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27) and second entries (Rt in Fig.8, see also Col.9 line 25 – Col.10 line 20 and Col.11 line 64 – Col.12 line 27) in the register file.

40. Regarding claim 33, Barry has taught an execution unit in a microprocessor, the execution unit comprising:

a. Means for receiving a plurality of numbers (see An/Rz of Fig.6). Here, the L2TBL instruction, which is the LTBL instruction modified to perform two look-up table look-ups (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.10 line 62 – Col.11 line 32).

b. Means for partitioning look-up memory into a plurality of look-up tables (see Col.7 lines 54-62 and Col.9 lines 63-67),

c. Means for looking up simultaneously a plurality of elements from the plurality of look-up tables (431/433 of Fig.4), each of the plurality of elements being in one of the plurality of look-up tables and being pointed to by one of the plurality of

numbers (see Col.9 lines 41-67 and Col.12 lines 14-27). Here, the above two pointers received point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

d. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32).

41. Regarding claim 34, Barry has taught an execution unit as in claim 33, wherein the means for receiving a plurality of numbers comprises:

a. Means for partitioning a string of bits into a plurality of segments to generate the plurality of numbers (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register address.

42. Regarding claim 35, Barry has taught an execution unit as in claim 34, wherein the single instruction specifies format information in which the plurality of numbers are stored in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register (see Fig.6A), and formats them into two segments of data, Rze and Rzo based on the even and odd source register address (see Col.10 line 62 – Col.11 line 32).

43. Regarding claim 37, Barry has taught an execution unit as in claim 34, wherein the string of bits is received from an entry of a register file (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register (see Fig.6A), and formats them

into two segments of data, Rze and Rzo based on the even and odd source register address (see Col.10 line 62 – Col.11 line 32).

44. Regarding claim 38, Barry has taught an execution unit as in claim 37, wherein the single instruction specifies an index of the entry (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register (see Fig.6A), and formats them into two segments of data, Rze and Rzo based on the even and odd source register address (see Col.10 line 62 – Col.11 line 32).

45. Regarding claim 39, Barry has taught an execution unit as in claim 33, further comprising:

a. Means for storing the plurality of elements in an entry of a register file (see Col.10 line 62 – Col.11 line 32).

46. Regarding claim 40, Barry has taught an execution unit as in claim 39, wherein the single instruction specifies an index of the entry (see Rt of Fig.6A and Col.10 line 62 – Col.11 line 32).

47. Regarding claim 41, Barry has taught an execution unit as in claim 39, wherein the single instruction specifies format information in which the plurality of elements are stored in the entry (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies that the plurality of data entries are stored at even and odd addresses into the register file, thus specifying the format.

48. Regarding claim 43, Barry has taught an execution unit as in claim 33, wherein the single instruction specifies a total number of entries contained in each of the plurality of look-up tables (see Col.10 line 62 - Col.11 line 48). Here, Barry supports three instructions (STBL, S2TBL and S4TBL) that each specify a different number of entries in the look-up tables, as well as the size of each entry.

49. Regarding claim 44, Barry has taught an execution unit as in claim 44, wherein the total number of entries is one of:

a. 256 (see Col.11 line 44),

b. 512,

c. 1024.

50. Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 44.

51. Regarding claim 45, Barry has taught an execution unit as in claim 33, wherein the single instruction specifies a total number of bits used by each entry contained in the plurality of look-up tables (see Col.12 lines 14-28). Here, the "size" field of the S2TBL instruction (see Fig.8A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

52. Regarding claim 46, Barry has taught an execution unit as in claim 45, wherein the total number of bits is one of:

a. 8 ("two bytes" in Col.12 lines 14-28),

b. 16 ("two halfwords" in Col.12 lines 14-28),

c. 24.

53. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 46.

54. Regarding claim 47, Barry has taught an execution unit in a microprocessor, the execution unit comprising:

a. Means for receiving a string of bits (see Col.10 line 62 – Col.11 line 32),

b. Means for generating a plurality of indices using a plurality of segments of bits in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables.

c. Means for look up simultaneously a plurality of entries from a plurality of look-up tables (431/433 of Fig.4) using the plurality of indices (see Col.9 lines 41-67, Col.11 lines 1-13 and Col.12 lines 14-27). Here, the above two indices point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

d. Means for combining the plurality of entries into a first result (see Col.11 lines 10-32). Here, the two outputted entries from each look-up table are combined and stored in register Rt, with each entry stored in one half of register Rt.

e. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32).

55. Regarding claim 48, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for receiving a plurality of data elements specifying the plurality of segments in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register addresses.

56. Regarding claim 49, Barry has taught an execution unit as in claim 48, wherein the plurality of data elements are received from an entry in a register file (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register addresses.

57. Regarding claim 50, Barry has taught an execution unit as in claim 49, wherein the single instruction specifies an index of the entry in the register file (see Col.12 lines 14-28). The L2TBL instruction specifies the register indices (see Fig.6A) of two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.11 lines 10-32).

58. Regarding claim 51, Barry has taught an execution unit as in claim 48, further comprising:

a. Means for receiving a bit pointer, wherein the plurality of segments in the string of bits are determined using the bit pointer and the plurality of data elements (see

Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register address. Thus, the L2TBL instruction can be considered a bit pointer.

59. Regarding claim 52, Barry has taught an execution unit as in claim 51, further comprising:

a. Means for generating a new bit pointer using the first result (see Col.10 line 62 – Col.11 line 32). Here, because the L2TBL instruction both reads and writes to the register file, when the first result is written into the register file, it is read on subsequent instructions, thus creating a new bit pointer for those instructions to use.

60. Regarding claim 53, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for receiving an offset, wherein the plurality of indices are determined using the offset and the plurality of segments of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables.

61. Regarding claim 54, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for partitioning look-up memory into the plurality of look-up tables before said looking-up (see Col.7 lines 54-62 and Col.9 lines 63-67 and Col.11 lines 10-13).

62. Regarding claim 55, Barry has taught an execution unit as in claim 54, wherein the look-up memory comprises a plurality of look-up units, and wherein the means for partitioning look-up memory comprises:

a. Means for configuring the plurality of look-up units into the plurality of look-up tables (see Col.7 lines 54-62 and Col.9 lines 63-67 and Col.11 lines 10-13).

63. Regarding claim 56, Barry has taught an execution unit as in claim 55, wherein each of the plurality of look-up units comprises 256 8-bit entries (see Col.10 line 62 - Col.11 line 48). Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry.

64. Regarding claim 61, Barry has taught an execution unit as in claim 55, wherein the plurality of look-up tables are configured according to an indicator in an entry in a register file (see Col.11 lines 11-32). Here, the data type and the instruction type define how the look-up tables will be configured, and because the data type is determined from the data in the register file, it can be considered an indicator in the register file.

65. Regarding claim 62, Barry has taught an execution unit as in claim 61, wherein the single instruction specifies an index of the entry in the register file (see Col.12 lines 14-28). The S2TBL instruction specifies the register indices (see Fig.8A) of two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at

Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

66. Regarding claim 63, Barry has taught an execution unit as in claim 47, wherein the means for combining the plurality of entries comprises:

a. Means for selecting valid data from the plurality of entries (see Col.11 lines 10-27).

67. Regarding claim 64, Barry has taught an execution unit as in claim 63, further comprising:

a. Means for generating an indicator indicating whether none of the plurality of entries is valid (see Col.11 lines 5-9). Here, the results being indeterminate is considered to indicate that none of the entries were of a valid data type.

68. Regarding claim 65, Barry has taught an execution unit as in claim 63, wherein the valid data is selected according to priorities of the look-up tables from which the plurality of entries are looked up (see Col.11 lines 10-32). Here, the "priorities" of the look-up tables are considered to be those entries that are valid have priority over non-valid, indeterminate entries (see Col.11 liens 5-9). Thus, valid entries are determined using "priorities" of the look-up tables.

69. Regarding claim 66, Barry has taught an execution unit as in claim 63, wherein the means for combining the plurality of entries further comprises:

a. Means for formatting the valid data according to a type of the valid data (see Col.11 lines 10-27). Here, the data is further formatted by sign extension if it is of types "dual half-word" or "dual-byte", and not sign extended if it's of the other valid type ("dual word").

70. Regarding claim 67, Barry has taught an execution unit as in claim 66, wherein the type of the valid data is one of:

a. Zero fill,

b. Signed magnitude (see Col.11 lines 20-27),

c. Two's complement.

71. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 67.

72. Regarding claim 68, Barry has taught an execution unit as in claim 67, further comprising:

a. Means for retrieving a sign bit from the string of bits for the valid data, wherein the first result is obtained by formatting the valid data using the sign bit when the type of the valid data is signed magnitude (see Col.11 lines 20-27). Here, when data of a valid type is to be sign extended, it is formatted to the correct size by sign-extending the sign bit.

73. Regarding claim 69, Barry has taught an execution unit as in claim 47, wherein an entry in the plurality of entries contains:

a. Information indicating whether the entry is valid (see Col.11 lines 10-27). Here, "double-word" data is not supported by the L2TBL instruction, and thus an unsupported data type in an entry of the look-up table inherently indicates that it is not valid, and a supported data type indicates that it is valid.

b. Information indicating a type of the entry (see Col.11 lines 10-27). Again, data of a supported type in an entry of the look-up table inherently is an indication of the type of the data in the entry.

c. Information indicating a number of bits of a code word to be decoded (see Col.11 lines 10-27). Here, the type of data defines the length of the data that is to be read out of the look-up table.

74. Regarding claim 70, Barry has taught an execution unit as in claim 47, wherein the string of bits is received from an entry in a register file (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables.

75. Regarding claim 71, Barry has taught an execution unit as in claim 70, wherein the single instruction specifies an index of the entry in the register file (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo.

76. Regarding claim 72, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for receiving a first number indicating a position of a last bit of input in the string of bits (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and

Col.12 lines 14-28). Thus, the size of the data indicates where the last bit of the input string will be in relation to the register size.

77. Regarding claim 74, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code (see Col.15 lines 13-17 and Col.15 line 63 – Col.16 line 4).

78. Regarding claim 75, Barry has taught an execution unit as in claim 74, wherein the predetermined code represents an end of block condition (see Col.15 lines 13-17 and Col.15 line 63 – Col.16 line 4).

79. Regarding claim 76, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for receiving at least one format (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

b. Means for formatting the string of bits into at least one escape data according to the at least one format (see Col.11 lines 10-32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two

look-up tables. The output for the look-up tables at these indices is formatted, i.e. sign-extended, if necessary according to the "size" field of the instruction.

c. Means for combining the at least one escape data and the first result into a second result. (see Col.10 line 62 – Col.11 line 32). Here, the two outputted entries from each look-up table are sign-extended (i.e. combined with the escape data) if necessary, and then combined and stored in register Rt, with each entry stored in one half of register Rt.

80. Regarding claim 77, Barry has taught an execution unit as in claim 76, wherein one of the at least one format is for data of a type which is one of:

a. Zero fill,

b. Signed magnitude (see Col.11 lines 20-27),

c. Two's complement.

81. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 77.

82. Regarding claim 78, Barry has taught an execution unit as in claim 76, wherein the at least one format is received from an entry of a register file (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28). Here, the format is specifies in the "size" field of the instruction, and then the actual format (data of a certain data type) is read out from an entry in the register file.

83. Regarding claim 79, Barry has taught an execution unit as in claim 78, wherein the single instruction specifies an index of the entry in the register file (see Col.11 lines 10-32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two

segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The output for the look-up tables at these indices is formatted, i.e. sign-extended, if necessary according to the "size" field of the instruction.

84. Regarding claim 80, Barry has taught an execution unit as in claim 1, wherein the look-up memory comprises a plurality of look-up units, and wherein the first circuit is to configure the plurality of look-up units into a third plurality of look-up tables in response to the microprocessor receiving a third instruction (see "LTBL" on Col.10 lines 24-61). Here, the LTBL instruction partitions the look-up memory into a single look-up table (see Col.10 lines 24-61).

85. Regarding claim 81, Barry has taught an execution unit as in claim 4, wherein each of the third plurality of look-up units contains 256 8-bit entries (see Col.10 line 62 - Col.11 line 48). Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry.

86. Regarding claim 82, Barry has taught an execution unit as in claim 33, wherein the single instruction specifies a total number of entries contained in each of the plurality of look-up tables (see Col.10 line 62 - Col.11 line 48). Here, Barry supports three instructions (STBL, S2TBL and S4TBL) that each specify a different number of entries in the look-up tables, as well as the size of each entry.

87. Regarding claim 83, Barry has taught an execution unit as in claim 4, wherein a total number of entries in each of the third plurality of look-up tables is one of:

a. 256 (see Col.11 line 44),

b. 512,

c. 1024.

88. Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 6.

89. Regarding claim 84, Barry has taught an execution unit as in claim 33, wherein the single instruction specifies a total number of bits used by each entry contained in the plurality of look-up tables (see Col.12 lines 14-28). Here, the "size" field of the S2TBL instruction (see Fig.8A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

90. Regarding claim 85, Barry has taught an execution unit as in claim 4, wherein a total number of bits used by each entry in the third plurality of look-up tables is one of:

a. 8 ("two bytes" in Col.12 lines 14-28),

b. 16 ("two halfwords" in Col.12 lines 14-28),

c. 24.

91. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 7.

Claim Rejections - 35 USC § 103

92. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

93. Claims 16, 18, and 27-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Barry et al., U.S. Patent No. 6,397,324 (herein referred to as Barry) in view of Priem, U.S. Patent Number 5,768,628 (herein referred to as Priem).

94. Regarding claim 16, Barry has taught an execution unit in a microprocessor (Barry Fig.4), the execution unit comprising:

a. A plurality of look-up tables (Barry 431/433 of Fig.4),

b. A first circuit coupled to the plurality of look-up tables, the first circuit, in response to the microprocessor receiving a single instruction, replacing at least one entry in at least one of the plurality of look-up tables with at least one data element (Barry Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (Barry Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (Barry Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (Barry Col.10 lines 5-20).

95. Barry has not taught using a Direct Memory Access (DMA) controller. Priem has taught using a Direct Memory Access (DMA) controller (Priem Abstract; column 6, line 45 to column 7, line 35; column 7, line 56 to column 8, line 4; Figure 3; and Figure 4). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory controller allows data transfers to occur as quickly as possible (Priem column 6, lines 52-54), thereby increasing processor speed and efficiency (Priem column 2, line 64 to column 3, line 5). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the memory controller of Priem in the device of Barry to improves processor speed and efficiency.

96. Regarding claim 18, Barry has taught an execution unit in a microprocessor (Barry Fig.4), the execution unit comprising:

a. A plurality of look-up tables (Barry 431/433 of Fig.4),

b. A first circuit coupled to the plurality of look-up tables, the first circuit, in response to the microprocessor receiving a single instruction, replacing at least one entry for each of the plurality of look-up tables with a plurality of data elements (Barry Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (Barry Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (Barry Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register

file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (Barry Col.10 lines 5-20).

97. Barry has not taught using a Direct Memory Access (DMA) controller. Priem has taught using a Direct Memory Access (DMA) controller (Priem Abstract; column 6, line 45 to column 7, line 35; column 7, line 56 to column 8, line 4; Figure 3; and Figure 4). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory controller allows data transfers to occur as quickly as possible (Priem column 6, lines 52-54), thereby increasing processor speed and efficiency (Priem column 2, line 64 to column 3, line 5). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the memory controller of Priem in the device of Barry to improves processor speed and efficiency.

98. Regarding claim 26, Barry has taught an execution unit in a microprocessor (Barry Fig.4), the execution unit comprising:

a. Means for replacing at least one entry in at least one of a plurality of look-up units (Barry 431/433 of Fig.4) in a microprocessor unit with at least one number (Barry Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (Barry Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (Barry Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and

Rto that will be written into the look-up table entries pointed to by the pointers created (Barry Col.10 lines 5-20).

b. Wherein the above means operate in response to the microprocessor receiving a single instruction (Barry "S2TBL" on Col.12 lines 13-27).

99. Barry has not taught using a Direct Memory Access (DMA) controller. Priem has taught using a Direct Memory Access (DMA) controller (Priem Abstract; column 6, line 45 to column 7, line 35; column 7, line 56 to column 8, line 4; Figure 3; and Figure 4). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory controller allows data transfers to occur as quickly as possible (Priem column 6, lines 52-54), thereby increasing processor speed and efficiency (Priem column 2, line 64 to column 3, line 5). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the memory controller of Priem in the device of Barry to improves processor speed and efficiency.

100. Regarding claim 27, Barry has taught an execution unit in a microprocessor (Barry Fig.4), the execution unit comprising:

a. Means for replacing at least one entry for each of a plurality of look-up units in a microprocessor with a plurality of numbers (Barry Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (Barry Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (Barry Col.12 lines 14-27). The S2TBL instruction further

specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (Barry Col.10 lines 5-20).

b. Wherein the above means operate in response to the microprocessor receiving a single instruction (Barry "S2TBL" on Col.12 lines 13-27).

101. Barry has not taught using a Direct Memory Access (DMA) controller. Priem has taught using a Direct Memory Access (DMA) controller (Priem Abstract; column 6, line 45 to column 7, line 35; column 7, line 56 to column 8, line 4; Figure 3; and Figure 4). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory controller allows data transfers to occur as quickly as possible (Priem column 6, lines 52-54), thereby increasing processor speed and efficiency (Priem column 2, line 64 to column 3, line 5). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the memory controller of Priem in the device of Barry to improves processor speed and efficiency.

102. Regarding claim 28, Barry has taught an execution unit as in claim 27, wherein a single index encoded in the instruction specifies a location of the at least one entry in the plurality of look-up units (Barry Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (Barry Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables

(Barry Col.12 lines 14-27). Thus, each instruction contains an encoded pointer that is an index into a corresponding look-up table.

103. Regarding claim 29, Barry has taught an execution unit as in claim 27, wherein a single index encoded in the instruction specifies a total number of the at least one entry in the plurality of look-up units (Barry Col.10 line 62 - Col.11 line 48). Here, Barry supports three instructions (STBL, S2TBL and S4TBL) that each specify a different number of entries in the look-up tables based on the instruction itself.

104. Regarding claim 30, Barry has taught an execution unit as in claim 27, wherein a source address of the plurality of numbers is specified in an entry of a register file (Barry Col.12 lines 14-28). The S2TBL instruction specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (Barry Col.10 lines 5-20). Thus because the index of the register file is specified in the instruction, and the source address of the plurality of numbers is the index of that entry in the register file, the source address of the plurality of numbers is specified in an entry in the register file.

105. Regarding claim 31, Barry has taught an execution nit as in claim 30, wherein the single instruction specifies an index of the entry in the register file (Barry Col.12 lines 14-28). The S2TBL instruction specifies the register indices (Barry Fig.8A) of two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (Barry Col.10 lines 5-20).

106. Regarding claim 32, Barry has taught an execution unit as in claim 27, wherein an identity number encoded in the single instruction specifies a memory controller (Barry Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers to an entry in one of the look-up tables (see Col.12 lines 14-27). Because the pointers are "encoded" in the instruction, and because all accesses to the look-up tables use a memory controller (see Col.7 lines 50-62), the instruction inherently specifies the memory controller. Barry has not taught the memory controller is a Direct Memory Access (DMA) controller. Priem has taught a Direct Memory Access (DMA) controller (Priem Abstract; column 6, line 45 to column 7, line 35; column 7, line 56 to column 8, line 4; Figure 3; and Figure 4). A person of ordinary skill in the art at the time the invention was made would have recognized that the memory controller allows data transfers to occur as quickly as possible (Priem column 6, lines 52-54), thereby increasing processor speed and efficiency (Priem column 2, line 64 to column 3, line 5). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the memory controller of Priem in the device of Barry to improves processor speed and efficiency.

Response to Arguments

107. Applicant's arguments, see Amendment, filed 13 December 2004, with respect to the rejection(s)of claim(s) 16, 18, 26, and 27 have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of the above.

108. Applicant's arguments, see Amendment, filed 13 December 2004, with respect to the rejection(s)of claim(s) 73 have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of above.

109. Applicant's arguments filed 13 December 2004 have been fully considered but they are not persuasive.

110. Applicant argues in essence on page 19 regarding the obviousness-type double patenting rejection "Each of the claims...specifically recites a particular arrange of *an execution unit* in a microprocessor. Such arrangements for *an execution unit* in a microprocessor are not obvious in view of the methods recited in...the copending Application..." This has not been found persuasive. Claims 14, 16, 18, 23, 26, 27, and 33 are mainly claiming the function of an execution unit. The arrangement of the execution unit, i.e. the hardware in the unit, is inherent to the functions being described in regards to the plurality of look-up tables, the memory controller, etc. The method claimed in the copending application also claim the functionality. As can be seen by the tables in the rejection above, the body of the claims are almost exactly the same.

111. Applicant argues in essence on pages 20-21, 22-23, 24, and 32-33 "...Barry does not show '***the first circuit***' partitioning the look-up memory into different look-up table configurations. Barry shows the designer's choice of split memories into separate banks which are addressable independently." This has not been found persuasive. The claim mainly describes how the first circuit functions, which is what is taught by Barry, not the hardware. The circuitry the argument alludes to is more inherent to the claim as necessary for the claim to function as claimed. The circuit in Figure 1 does the function described in the claim by

executing the instructions referred to in the citation above, so Figure one could be considered a circuit. Barry has taught an instruction that is executed in a circuit, and the instruction causes the circuit to perform functions that partitions look-up memory. The instruction partitions the memory into either one, two, or four areas, as described by Barry in column 10, lines 23-28. The example of the L2TBL instruction makes the partitioning more apparent in column 11, lines 10-13 by stating that the memory banks are treated as individual, in dependent banks, i.e. separate look-up memories. This is not a designer choice, since it is the instruction that tells the execution unit to partition the memories not the designer. Please see the attached dictionary definition of "partition".

112. Applicant argues in essence on page 21 and 33 "...memory banks 431/433 of Figure 4 of Barry are not host memory; and memory interface unit of 485 of Figure 4 of Barry is not a memory controller for controlling host memory." This has not been found persuasive. Host memory is just memory that is used independently. Memory banks 431/433 of Figure 4 are host memory according to this definition. The memory interface unit 485 of Figure 4 controls the memory accesses. Please see the attached dictionary definition of "host".

113. Applicant argues in essence on page 24 "...Barry does not show any execution unit receiving data from Rz register as a string of bits." This has not been found persuasive. Computers are binary and use 1's and 0's to communicate within the system. Strings of bits, i.e. the multiple 1's and 0's used by the system to represent data, are passed between elements in the system. In Barry, the L2TBL instruction uses half-words which are passed between the elements in the system. These half-words are strings of bits.

114. Applicant argues in essence on page 24-25

Barry does not shown an execution unit that has a 'fifth circuit' to format the string of bits into at least one escape data using the at least on format and a 'sixth circuit' to combine the at least one escape data with the first result into a second result

115. This has not been found persuasive. The L2TBL instruction specifies a particular format with the sign-extension bit. The sign-extension bit designates whether the output data is sign-extended or not. The instruction retrieves data into the Rt register, and, depending on the sign-extension bit, sign-extends the data or not. The instruction only retrieves half-word data which is loaded into the H0 portion of the register. The sign-extension bit determines whether a string of bits needs to be formatted for the H1 portion of the registers, i.e. formatting escape data. The H1 portion is then combined with the H0 portion to make a sign-extended result in Rz (Barry column 11, lines 15-27).

116. Applicant argues in essence on page 25 "Barry does not show *an entry of the register file* that specifies 'the actual format.' This has not been found persuasive. The claims states "wherein the at least one format is received from an entry of a register file." The format is specified by the sign-extension field, which specifies whether the data in sign-extended format or not. The sign-extension bit is part of the instruction currently being executed, which is stored in an instruction register. The instruction register is a single register register file. The format comes from the instruction in the instruction register.

117. Applicant argues in essence on page 26 "The sign extension bit specifies the information need for the instruction of Barry regarding the format. No further information form an entry in

the register file is need for the instruction of Barry." This has not been found persuasive. The Rz register provides the index not the sign extension bit.

118. Applicant argues in essence on page 26 "...there would be no 'means for receiving a plurality of data elements specifying the plurality of segments...' in any execution unit of Barry." This has not been found persuasive. The Rz register is partitioned in to a plurality of segments that are the indices. The indices are data elements that are half-words.

119. Applicant argues in essence on pages 26-27 "Applicant respectfully submits that it is improper to consider the data in Rze and Rzo as both the 'string of bits' and 'the plurality of data elements'." This has not been found persuasive. Applicant does not explain why this is improper. The Examiner can merely reiterate that the Rz register in the register file is partitioned into a plurality of segments that are the indices, which are the data elements.

120. Applicant argues on page 27 in essence "...Since the L2TBL instruction does not encode any information about a pointer specified in the claim, the L2TBL instruction cannot be considered as corresponding to the bit pointer recited in the claim." This has not been found persuasive. The indices are pointers. They are logical pointers that, when added to the base in An, point the location of the data being retrieved from the table memory. A pointer points to a location in memory.

121. Applicant argues in essence on page 28 "...There is no process of 'selecting' in Barry...Nothing in Barry corresponds to 'means for selecting...'." This has not been found persuasive. When the data is retrieved from the data table, it is selecting data from the table, since it is choosing one, two, or four entries out of the all the entries in the tables, and the data is

always seen as valid. There is nothing in Barry to indicate that that data is never considered valid.

122. Applicant argues in essence on page 28 ""The size field 23-22 does not indicate the last bit of the input of string bits." This has not been found persuasive. The size filed does indicate the position of the last bit. It determines whether the size is a half-word, so that the last bit is at the half-word location at the end of the H0 location, or whether it is size-extended, which means that the last bit of the data is located at the end of the H1 location.

123. Applicant argues in essence on page 29 "...Barry does not have, in an execution unit, 'means for generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code', such as a code for an end of block condition." This has not bee found persuasive. Barry states in column 11, lines 13-14 that the maximum architecture size of the look-up table is 64 K entries. This means that the indices cannot surpass the 64 K entry limit, i.e. there is an end limit to the indices, which is an end of block condition.

Conclusion

124. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as follows. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

a. Priem, U.S. Patent Number 5,968,148, has taught a DMA controller that loads and stores table information to and from main memory to local memory.

125. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:30am-5:00pm.

126. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

127. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
4 March 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100